8.0 - 1 0.2 1 4

T79-10994 NNT
NASA CR-
160649

JSC- 13945

PROGRAM DOCUMENTATION
"AS-BUILT" DESIGN SPECIFICATION
FOR
GENERALIZED LINEAR MODEL ANALYSIS OF VARIANCE
PROGRAM (GLMAOV)

JOB ORDER 71-593

Prepared By
Lockheed Electronics Company, Inc.
Systems and Services Division
Houston, Texas

Contract NAS 9-15200
For
EARTH OBSERVATIONS DIVISION
SCIENCE AND APPLICATIONS DIRECTORATE

National Aeronautics and Space Administration
LYNDON B. JOHNSON SPACE CENTER
Houston, Texas

March 1978

LEC- 12085

JSC- 13945

PROGRAM DOCUMENTATION

"AS-BUILT" DESIGN SPECIFICATION FOR GENERALIZED LINEAR MODEL ANALYSIS OF VARIANCE PROGRAM (GLMAOV)

Job Order 71-593

(TIRF 77-0042)

Prepared by
Jeannie S. Gor

APPROVED BY

for James A Wilkinson

Philip L. Krumm, Acting Supervisor
Scientific Applications Section

Lockheed Electronics Company, Inc.
Systems and Services Division
Houston, Texas

March 1978

LEC- 12085

TABLE OF CONTENTS

1. SCOPE

This specification establishes the design for a generalized linear model analysis of variance (AOV) computer program.

The requirements specification for this program was provided by the Research, Test, and Evaluation (RT&E) Branch of the Earth Observations Division (EOD) of The National Aeronautics and Space Administration/Lyndon B. Johnson Space Center (NASA/JSC).

2. APPLICABLE DOCUMENTS

The following documents, of exact issue shown, form a part of this specification to the extent specified herein:

- "Generalized Linear Model AOV Program Requirements," provided by M. D. Pore/LEC.
- Task Agreement 77-4, Job Order 81-127
- TIRF 77-0042

3. SYSTEM DESCRIPTION

3.1 HARDWARE DESCRIPTION

Not applicable.

3.2 SOFTWARE DESCRIPTION

The purpose of the program GLMAOV is to implement an analysis of variance for experimental designs. The program is to complete an analysis of variance for unbalanced designs, designs with missing data and designs with multiple solutions. The program is in a general form and includes analyses of covariance and regression. Pseudo-inverses of the model matrices required in the analyses will be computed using an algorithm identified as "Greville's Method."

The program is coded in the Univac 1108/EXEC 2 Fortran V language but using only Fortran IV-G capability for future implementation on the IBM/370.

3.2.1 SOFTWARE COMPONENT NO. 1 (GLMAOV)

GLMAOV functions as a driver program for the generalized Linear Model Analysis of Variance Program (LMAOV). It reads in the number of analyses to be taken, the dimensions of the input math models for each analysis and then computes base addresses for arrays used in LMAOV. The general linear model used is:

$$Y = X\beta + e$$

with β subjected to the constraint:

$$R\beta = t$$

The hypothesis tested is

$$H_o: \lambda\beta = h$$

against the alternative

$$H_1: \ \lambda\beta \neq h$$

3.2.1.1 Linkages

GLMAOV is not referenced by any other program. It calls subroutine LMAOV.

3.2.1.2 Interfaces

Interface between GLMAOV and LMAOV is done via the calling arguments of LMAOV.

3.2.1.3 Inputs

GLMAOV requires an input card deck as follows:

[Right justify all numbers]

	Column	Description
1st card	1-3	Number of analyses to be taken
	5-6	NP (Row dimension of Y and X)
	8-9	NQ (Column dimension of X, R, and λ)
	11-12	NM (Row dimension of R and t)
	14-15	NN (Row dimension of λ and h)
	17-26	TOLENC (Tolerance Level)

Subsequent Cards [Required inputs for first analysis for subroutine LMAOV; see Software Component 2 (LMAOV), section 3.2.2]

1-2 NEW NN (Row dimension of next set of λ and L)

[Inputs for second analysis for subroutine LMAOV]

<u>Column</u>	<u>Description</u>
[Inputs for third analysis for subroutine LMAOV]	

●

●

●

●

●

●

●

End-of-File card

3.2.1.4 Outputs

If the math models used are found to be too large dimensionally for the space allocated, GLMAOV outputs this message and terminates:

****** Dimensions of math models are too large for use in this program ** Storage capacity of GLMAOV is exceeded.

3.2.1.5 Storage Requirements

To be determined.

3.2.1.6 Description

GLMAOV first reads in the number of analyses to be taken, the dimensions of the math models [See 3.2.2 SOFTWARE COMPONENT 2 (LMAOV)] and an input tolerance level. If TOLENC is not input, TOLENC defaults to .0001. GLMAOV then computes base addresses of all arrays used in LMAOV from the dimensions that were read. GLMAOV has a large working array of size 42,000 which will be divided using the base addresses that were computed. These base addresses are passed to LMAOV via the calling arguments as starting addresses of working arrays for LMAOV. The call for LMAOV is in a DO-loop from 1 to number of analyses. A dimension for NN must be input before each analysis whether NN will be changed or not.

3.2.1.7 Flow Chart

Not applicable.

3.2.1.8 Listing

See Appendix A

3.2.2 SOFTWARE COMPONENT NO. 2 (LMAOV)

LMAOV is a generalized Linear Model Analysis of Variance Program for experimental designs. It will complete the analysis for unbalanced designs, designs with missing data and designs with multiple solutions. The model is written in a general form and includes analyses of covariance and regression. The general model used is:

$$Y = X\beta + e$$

with β subjected to the constraint:

$$R\beta = t$$

The hypothesis tested is

$$H_o: \quad \lambda\beta = h$$

against the alternative

$$H_1: \quad \lambda\beta \neq h$$

3.2.2.1 Linkages

LMAOV is called from GLMAOV and will reference TRANSP, MULTMX, SUBMX and PSINV.

3.2.2.2 Interfaces

Interfaces provided by calling sequences.

3.2.2.3 Inputs

Calling Sequence:

Call LMAOV (YVEC, XMX, BVEC, RMX, CTVEC, RINV, TVEC, GMX, GINV, HINV, CHV, HVEC, WKR, XTRP, HMX, GWKR, CHVEC, RWKR, CMX, CHMX, WKRS, CINV, MMX, ZWRK, ZWST, PWRK, MINV, PWST, ZTRP, PPWRK, EWRK, WORKST, NP, NQ, NM, NN, NALYS, TOLENC, NWS)

Parameter	Dimensions	In/Out	Description
YVEC	NP	In	Random vector of observations (Y vector)
XMX	(NP, NQ)	In	Known matrix that includes the design and regression matrices (X matrix)
BVEC	NQ	In	Vector of random and non-random parameters including higher power terms and interaction terms (β vector). The test variables, regression variables and con-commitant variables are subsets of β
RMX	(NM, NQ)	In	R matrix
CTVEC	(NM)	In	Working array
RINV	(NQ, NM)	In	Working array
TVEC	(NM)	In	t vector
GMX	(NN, NQ)	In	λ matrix
GINV	(NQ, NN)	In	Working array
HINV	(NQ, NN)	In	Working array
CHV	(NQ)	In	Working array
HVEC	(NN)	In	h vector

Parameter	Dimension	In/Out	Description
WKR	(NQ, NQ)	In	Working array
XTRP	(NQ, NP)	In	Working array
HMX	(NN, NQ)	In	Working array
GWKR	(NN, NN)	In	Working array
CHVEC	(NN)	In	Working array
RWKR	(NM, NM)	In	Working array
CMX	(NQ, NQ)	In	Working array
CHMX	(NN, NM)	In	Working array
WKRS	(NQ, NQ)	In	Working array
CINV	(NQ, NQ)	In	Working array
MMX	(NP, NQ)	In	Working array
ZWRK	(NP)	In	Working array
ZWST	(NP)	In	Working array
PWRK	(NQ, NP)	In	Working array
MINV	(NQ, NP)	In	Working array
PWST	(NP, NP)	In	Working array
ZTRP	(NP)	In	Working array
PPWRK	(NP, NP)	In	Working array
EWRK	(NQ, NQ)	In	Working array
WORKST	1	In	Starting address of the remaining storage of the large working array in GLMAOV
NP	1	In	Row dimension of Y and X
NQ	1	In	Column dimension of X, R, and λ

Parameter	Dimension	In/Out	Description
NM	1	In	Row dimension of R and t
NN	1	In	Row dimension of λ and h
NALYS	1	In	Number of analyses to be taken
TOLENC	1	In	User input tolerance level
NWS	1	In	Last computed base address from GLMAOV

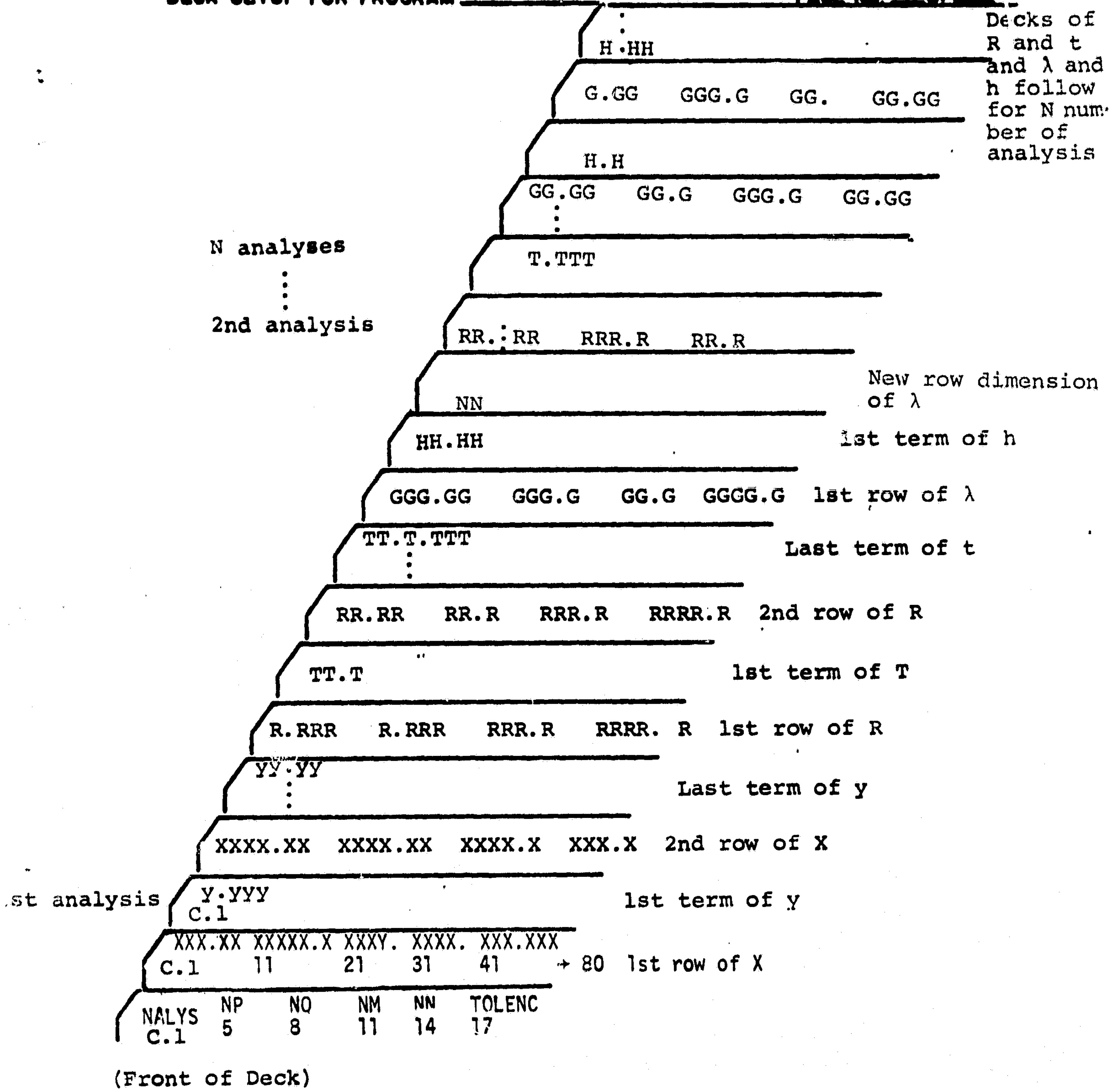
Input of the elements of the known vectors and matrices are read from cards in the following manner:

1. X(XMX) and Y(YVEC) are read simultaneously. First one row of X is read followed by the first element of Y. The second row of X is read followed by the second element of Y, etc. Hence NP lines of NQ + 1 elements.
2. R(RMX) and t(TVEC) are read in the same manner as (1.), therefore NM lines of NQ + 1 elements.
3. λ(GMX) and h(HVEC) are also read in the same manner as (1.) and (2.), hence NN lines of NQ + 1 terms.

For a row of terms, a maximum of 8 terms may be input per card, the first term in columns 1-10, second in columns 11-20, third in columns 21-30, etc. until (as for X) NQ terms are read from as many cards as needed for NQ terms. The card following will contain the first element of Y (columns 1-10) then the same process is repeated for the next row of X followed by the next term of Y and so on. Cards for R and t follow in the same manner; then cards for λ and h. If more than one analysis is performed, the data for R and t and the data for λ and h would follow in the same fashion as before. In the case that more than one analysis is taken, a card (punched in col. 1-2) is necessary that contains the row dimension (NN) of λ and L. This is done so that the number of rows of λ and L may vary from analysis to analysis. Even if the dimension remains the same with sucessive analysis, NN must be input with each set of inputs.

The user must use care in having the correction number of terms for each row for the correct number of rows. An example of the deck setup is located on the next page.

DECK SETUP FOR PROGRAM ________ PAGE NO. ___ OF ___


(Front of Deck)
NALYS C.1 NP 5 NQ 8 NM 11 NN 14 TOLENC 17
XXX.XX XXXXX.X XXXY. XXXX. XXX.XXX C.1 11 21 31 41 → 80 1st row of X
st analysis
Y.YYY C.1 1st term of y
XXXX.XX XXXX.XX XXXX.X XXX.X 2nd row of X
YY.YY Last term of y
R.RRR R.RRR RRR.R RRRR. R 1st row of R
TT.T 1st term of T
RR.RR RR.R RRR.R RRRR.R 2nd row of R
TT.T.TTT Last term of t
GGG.GG GGG.G GG.G GGGG.G 1st row of λ
HH.HH 1st term of h
NN New row dimension of λ
RR.RR RRR.R RR.R
T.TTT
GG.GG GG.G GGG.G GG.GG
H.H
G.GG GGG.G GG. GG.GG
H.HH
2nd analysis
N analyses
Decks of R and t and λ and h follow for N number of analysis

Also, input terms are presumed to be decimal (floating point) numbers of a maximum of 8 digits with a maximum of 6 digits to the right of the decimal and should be located within a card field of 10 columns, starting at column 1, i.e., 1-10, 11-20, 21-30, 31-40, ..., 71-80.

3.2.2.4 Outputs

The following messages are printed if computations described in Section 3.2.2.6 are found to be true:

1. * * * * * * * * * *
   REJECT H_O WITH PROB (TYPE I ERROR) = PROB (TYPE II ERROR) = 0.
   THE RESTRICTIONS AND HYPOTHESES ARE SEPARATELY CONSISTENT,
   BUT NOT JOINTLY COMPATIBLE.

2. * * * * * * * * * *
   H_O IS A LEGITIMATE HYPOTHESIS BUT THE MODEL IS NOT MATHEMATICALLY CONSISTENT.
   RECONSIDER THE ASSUMED RESTRICTIONS.

3. * * * * * * * * * *
   REJECT H_O WITH PROB (TYPE I ERROR) = PROB (TYPE II ERROR) = 0.
   THE ASSUMED RESTRICTIONS ARE MATEHMATICALLY CONSISTENT, BUT
   THE HYPOTHESIS IS NOT CONSISTENT.

4. * * * * * * * * * *
   REJECT H_O WITH PROB (TYPE I ERROR) = PROB (TYPE II ERROR) = 0.
   NEITHER THE ASSUMED RESTRICTIONS NOR THE HYPOTHESIS IS MATHEMATICALLY CONSISTENT.

5. * * * * * * * * * *
   THE HYPOTHESIS FUNCTION, β IS NOT ESTIMABLE. ANALYSIS CONTINUES.

If 1. through 4. of the above output is not printed, LMAOV prints an Analysis of Variance (AOV) table, the Best Linear Estimate (BLE) unrestricted with the covariance matrix and squared multiple correlation coefficient, the BLE restricted only by the model restrictions with the covariance matrix, and the BLE restricted by the hyposthesis with covariance matrix and squared multiple correlation coefficient. Refer to section 3.2.2.6 for definition of values of the AOV Table, covariance matrices, and squared multiple correlation coefficients.

AOV TABLE

Source of variation	Degree of freedom	Sum of squares	Mean squares	Value of F - statistic
Due to β	R_1	Q_3		
Due to β_N (Unadj.)	R_2	Q_2		
Due to β_H (ADJ)	$R_1 - R_2$	Q_1	MS1	$F = \frac{Q_1}{Q_o} \cdot \frac{NP - R_1}{R_1 - R_2}$
Error	$P - R_1$	Q_o	MS2	
Total	P	Q		

The Best Linear Estimate (BLE) unrestricted is:

$$[\hat{\Gamma}]$$

with covariance matrix:

$$[\mathrm{cov}\ (\hat{\Gamma})]$$

And squared multiple correlation coefficient:

$$\Gamma^2$$

The BLE restricted only by the model restricted is:

$$[\hat{\alpha}]$$

with covariance matrix

$$[\mathrm{cov}\ \hat{\alpha}]$$

The BLE restricted by the hypothesis is:

$$[\hat{\beta}]$$

with covariance matrix

$$[\mathrm{cov}\ \hat{\beta}]$$

And squared multiple correlation coefficient:

$$\beta^2$$

LMAOV also outputs an error message and terminates execution when storage is exceeeded:

* * * * * DIMENSIONS OF MATH MODELS ARE TOO LARGE FOR USE IN GLMAOV * * STORAGE CAPACITY OF LMAOV IS EXCEEDED * * *

3.2.2.5 STORAGE REQUIREMENTS

To be determined.

3.2.2.6 Description

Initially LMAOV reads input data. Y(YVEC) and X(XMX) are first read simultaneously in a nested DO-loop. Each row of input is a row of X followed by the respective element of Y so that there are NP rows of NQ + 1 terms. The same operation is done for R(RMX) and t(TVEC), hence NM rows of NQ + 1 terms, and λ(GMX) and h(HVEC), NN rows of NQ + 1 terms.

The computations in LMAOV require the subroutine PSINV which computes the Moore-Penrose pseudo-inverse of a matrix. Given any non-zero $[p \times q]$ matrix A, the pseudo-inverse is the unique matrix A^+. (Refer to section 3.2.3 SOFTWARE COMPONENT NO. 3 (PSINV)).

Initial computation:

$$H = \lambda(I - R^+R)$$

A test is made to determine if h(HVEC) and t(TVEC) are zero vectors. Comparison to zero is actually a test against an arbitrarily small value, greater than absolute zero. The test that is made is:

$$\Sigma(a_i{**}2) < (\text{TOLENC}{**}2) * \text{AMAX}\left(1,\Sigma(a_i{**}2)\right)$$

If true, the vector is considered to be a zero vector (if TOLENC is not input by the user, default TOLENC is .0001).

If both h and t are zero vectors, LMAOV computes values for the output AOV table. If either/both h and t are non-zero vectors, the following tests are made:

A test for $\lambda\lambda^{+}h = h$ and $RR^{+}t = t$ is made. Equality of vectors is determined by:

$$\text{ABS}\left(\Sigma\left((a_i - b_i){**}2\right)\right) < (\text{TOLENC}{**}2) * \text{AMAX}\left[1,\left(\Sigma(a_i{**}2) + \Sigma(b_i{**}2)\right)\right]$$

If the test is true, the vectors are considered equal.

1. If $(\lambda\lambda^{+}h = h)$ and $(RR^{+}t \neq t)$ the following message is written and execution on this set of data is terminated:

   * * * *

   H_O IS A LEGITIMATE HYPOTHESIS BUT THE MODEL IS NOT MATHEMATICALLY CONSISTENT. RECONSIDER THE ASSUMED RESTRICTIONS.

2. If $(\lambda\lambda^{+}h \neq h)$ and $(RR^{+}t = t)$ the following message is written and execution on this set of data is terminated:

   * * * *

   REJECT H_O WITH PROB (TYPE I ERROR) = PROB (TYPE II ERROR) = 0. THE ASSUMED RESTRICTIONS ARE MATHEMATICALLY CONSISTENT, BUT THE HYPOTHESIS IS NOT CONSISTENT.

3. If $(\lambda\lambda^+h \neq h)$ and $(RR^+t \neq t)$ the following message is written and execution on this set of data is terminated:

* * * * * *

REJECT H_O WITH PROB (TYPE I ERROR) = PROB (TYPE II ERROR) = 0. NEITHER THE ASSUMED RESTRICTIONS NOR THE HYPOTHESIS IS MATHEMATICALLY CONSISTENT.

4. If $(\lambda\lambda^+h = h)$ and $(RR^+t = t)$ another test is made:

$$HH^+(h - \lambda R^+t) = (h - \lambda R^+t)$$

If the above test is true, LMAOV conducts the test in 5. and computes the values for the AOV table; if not true, the following message is written and execution on this set of data is terminated:

* * * * * *

REJECT H_O WITH PROB (TYPE I ERROR) = PROB (TYPE II ERROR) = 0. THE RESTRICTIONS AND HYPOTHESES ARE SEPARATELY CONSISTENT, BUT NOT JOINTLY COMPATABLE.

5. If $H[X(I-R^+R)]^+ [X(I-R^+R)] \neq H$
then print:
* * * * * * * * * *

THE HYPOTHESIS FUNCTION, $\Lambda\beta$, IS NOT ESTIMABLE. ANALYSIS CONTINUES.

COMPUTATIONS FOR AOV TABLE AND BEST LINEAR ESTIMATES

Calculate:

$M = X(I - R^{+}R - H^{+}H)$

$C = (I - R^{+}R)X^{T}X(I - R^{+}R)$

$Z = Y - XR^{+}t - XH^{+}(h - \lambda R^{+}t)$

P = length of Y = NP

$R_1 = P - Tr(I - XC^{+}X^{T})$ rounded and truncated to an integer

$R_2 = R_1 - Tr(XC^{+}X^{T} - MM^{+})$ rounded and truncated to an integer

$Q_3 = Z^{T}XC^{+}X^{T}Z$

$Q_2 = Z^{T}MM^{+}Z$

$Q_1 = Z^{T}(XC^{+}X^{T} - MM^{+})Z$

$Q_o = Z^{T}(I - XC^{+}X^{T})_{Z}$

$Q = Z^{T}Z$

$$MS1 = \frac{Q_1}{R_1 - R_2}$$

$$MS2 = \frac{Q_o}{P - R_1}$$

$$F = \frac{Q_1}{Q_o} \cdot \frac{P - R_1}{R_1 - R_2}$$

The following AOV table is printed from the above computations.

Source of variation	Degree of freedom	Sum of squares	Mean squares	Value of F - statistic
Due to β	R_1	Q_3		
Due to β(N) (Unadj.)	R_2	Q_2		
Due to β(H) (Adj.)	$R_1 - R_2$	Q_1	MS1	F
Error	$P - R_1$	Q_o	MS2	
Total	P	Q		

The the following values are calculated:

$$YMEAN = \frac{\varepsilon Y(i)}{NP},\ i = 1 \text{ thru } NP$$

$$\hat{\Gamma} = X^+Y$$

$$COV(\hat{\Gamma}) = (X^TX)^+$$

$$\hat{\Gamma}R^2 = \frac{\hat{\Gamma}^TX^TY - NP\ (YMEAN)^2}{Y^TY - NP(YMEAN)^2}$$

$$\hat{\alpha} = R^+t + C^+X^T\ (Y - XR^+t)$$

$$COV(\hat{\alpha}) = C^+$$

$$\hat{\beta} = R^+t + H^+(h - \lambda R^+t) + M^+[Y - XR^+t - XH^+(h - \lambda R^+t)]$$

$$COV(\hat{\beta}) = (M^TM)^+$$

$$\hat{\beta}R^2 = \frac{[Z^T(M\hat{\beta}) - NP\ (YMEAN^2)]^2}{(Q - NP(YMEAN^2))\ (\hat{\beta}^TM^TM\ \hat{\beta}-NP(YMEAN^2))}$$

The above calculations are printed as such:

The Best Linear Estimate (BLSE) unrestricted is:

$$[\hat{\Gamma}]$$

with covariance matrix:

$$COV\ (\hat{\Gamma})$$

and squared multiple correlation coefficient:

$$[\hat{\Gamma}R^2]$$

The Best Linear Estimate restricted by the model restrictions is:

$$[\hat{\alpha}]$$

with covariance matrix:

$$\mathrm{COV}\ (\hat{\alpha})$$

The Best Linear Estimate restricted by the hypothesis is:

$$[\hat{\beta}]$$

with covariance matrix:

$$\mathrm{COV}\ (\hat{\beta})$$

and squared multiple correlation coefficient:

$$[\hat{\beta}R^2]$$

All transposition operations are done in subroutine TRANSP, matrix multiplication in subroutine MULTMX, and matrix subtraction in subroutine SUBMX.

All storage and working storage for LMAOV is passed by address into LMAOV via the calling argument from GLMAOV. This was done to provide the user with the maximum possible storage available for batch runs on the UNIVAC 1108. If the user exceeds this allocation of space, a message will be printed stating the violation. In which case the user should readjust the math model.

LMAOV uses the variable WORKST as the starting address of the remaining working space from ARRAY in GLMAOV. LMAOV will compute all base addresses for arrays in PSINV and divide WORKST appropriately. If storage space is exceeded, a message is written and the program stops execution on that particular set of data.

3.2.2.7 Flow Chart

Not applicable.

3.2.2.8 Listing

See Appendix A

3.2.3 SOFTWARE COMPONENT NO. 3 (PSINV)

The function of PSINV is to compute the Moore-Penrose pseudo inverse of a matrix. The defining properties of the pseudo-inverse are as follows: Given any non-zero $[pxq]$ matrix A, the pseudo-inverse is the unique matrix, denoted A^+, such that

1. $AA^+A = A$
2. $A^+AA^+ = A^+$
3. $(AA^+)^T = AA^+$
4. $(A^+A)^T = A^+A$

If A is a square and full rank matrix, then A has an inverse and $A^+ = A^{-1}$. If a is a non-zero $[px1]$ vector then

$$a^+ = (a^Ta)^{-1}a^T.$$

(A^T denotes the transpose of the matrix A.) The algorithm used for this computation is identified as "Greville's Method."

3.2.3.1 Linkages

Subroutine PSINV is called from LMAOV. PSINV calls TRANSP, MULTMX , and SUBMX.

3.2.3.2 Interfaces

Interface between PSINV and all other subroutines is via the calling arguments.

3.2.3.3 Inputs

Calling sequence is:

Call PSINV (AMX, OUTMX, NROW, NCOL, TOLENC, FCOL, TCOL, DK, CK, AK, DTR, BK, SCRACH)

Parameter	Dimension	In/Out	Description
AMX	(NROW, NCOL)	In	Real matrix from which pseudo-inverse will be taken
OUTMX	(NCOL, NROW)	Out	Real matrix which will be the pseudo-inverse matrix of AMX
NROW	1	In	Row dimension of AMX
NCOL	1	In	Column dimension of AMX
TOLENC	1	In	Tolerance level
FCOL	(NROW)	In	Working array
TCOL	(NROW)	In	Working array
DK	(NCOL)	In	Working array
CK	(NROW)	In	Working array
AK	(NROW)	In	Working array
DTR	(NCOL)	In	Working array
BK	(NROW)	In	Working array
SCRACH	(NCOL, NROW)	In	Working array

3.2.3.4 Outputs

PSINV outputs OUTMX matrix. OUTMX is the pseudo-inverse of AMX matrix.

3.2.3.5 Storage Requirements

To be determined.

3.2.3.6 Description

PSINV is a subroutine that uses 'Greville's method' as the algorithm to compute the pseudo-inverse of a matrix, denoted A^+. The matrix to be inverted and its dimensions are input parameters.

Computing the Pseudo-Inverse

Computation is accomplished by taking one column of the input matrix [AMX] at a time to produce one row of its inverse [OUTMX]. a_1 will be the first column of AMX.

If $\varepsilon(a_1{**}2) < (\text{TOLENC}{**}2) * \text{AMAX}(1,\varepsilon(a_1{**}2)$

then a_1 is considered equal to zero. (Test for zero vector).

If $a_1 = 0$ then $A_1^+ = 0$ (row vector, first row of OUTMX).

[TOLENC is input from LMAOV as a testing level for determining the closeness of the vector to a zero vector].

If $a_1 \neq 0$, then $A_1^+ = a_1^+ = (a_1^T a_1)^{-1} a_1^T$.

Note that $(a_1^T a_1)^{-1}$ results in matrix multiplication of (1,NN)X (NN,1) so that a (1X1) SCALAR results. Consequently the inverse is taken as 1./SCALAR.

AMX is then partitioned by columns in a DO-loop from K = 2, NCOL. A_K is the sub-matrix of AMX consisting of K columns and a_K is the K^{th} column.

$$A_K = \left[A_N, a_K\right]$$

$$N = K - 1$$

then d_K is computed as

$$d_K = A_N^+ a_K$$

then C_K is computed as

$$C_K = a_K - A_N d_K$$

then b_K is determined as

$$b_K = C_K^+, \text{ if } C_K \neq 0$$

or

$$b_K = \frac{d_K^T A^+{}_N}{1 + d_K^T d_K}, \text{ if } C_K = 0$$

Resulting in A_K^+ as

$$A_K^+ = \begin{bmatrix} A_N - d_K b_K \\ b_K \end{bmatrix}$$

b_K being an added row after each iteration until K = NCOL.

All transpositions are done by subroutine TRANSP, matrix multiplication by subroutine MULTMX, and matrix subtraction by subroutine SUBMX.

All working array storage is taken from already existing storage in LMAOV where base addresses are computed according to the dimensions of AMX and OUTMX.

3.2.3.7 Flow Chart

Not applicable.

3.2.3.8 Listing

See Appendix A

3.2.4 SOFTWARE COMPONENT NO. 4 (TRANSP)

The function of TRANSP is to transpose a matrix of real elements.

3.2.4.1 Linkages

TRANSP is referenced by LMAOV and PSINV. TRANSP does not require any other subroutines.

3.2.4.2 Interfaces

Interface between TRANSP and the two programs that call it are via the calling arguments of TRANSP.

3.2.4.3 Inputs

Calling Sequence:

Call TRANSP (A, B, M, N, MX, NX)

Parameter	Dimension	In/Out	Definition
A	(M, N)	In	The matrix to be transposed
B	(N, M)	Out	The transpose of A matrix
M	1	In	Row dimension of A
N	1	In	Column dimension of A
MX	1	In	First dimension of A as specified in the DIMENSION statement of the calling program

Parameter	Dimension	In/Out	Definition
NX	1	In	First dimension of B as specified in the DIMENSION statement of the calling program

3.2.4.4 Outputs

B matrix is output as the transpose of A.

3.2.4.5 Storage Requirements

To be determined.

3.2.4.6 Description

The transpose of A (M, N) matrix is B (N, M) matrix whose elements are:

$$B_{ji} = A_{ij}$$

3.2.4.7 Flow Chart

Not applicable.

3.2.4.8 Listing

See Appendix A

3.2.5 SOFTWARE COMPONENT NO. 5 (MULTMX)

MULTMX multiplies two matrices of type REAL.

3.2.5.1 Linkages

MULTMX is referenced by LMAOV and PSINV. MULTMX does not require any other subroutine.

3.2.5.2 Interfaces

Interface between MULTMX and the two subroutines that call it is via the calling arguments of MULTMX.

3.2.5.3 Inputs

Calling Sequence:

Call MULTMX (A, B, C, M, N, K, MX, NX)

Parameter	Dimension	In/Out	Definition
A	(M, N)	In	Two dimensional array containing elements of multiplicand matrix
B	(N, K)	In	Two dimensional array containing elements of multiplier matrix
C	(M, K)	Out	Two dimensional array containing product of A and B
M	1	In	First dimension of A and C
N	1	In	Second dimension of A and first dimension of B
K	1	In	Second dimension of B and C
MX	1	In	First dimension of A as specified by DIMENSION statement in the calling program
NX	1	In	First dimension of B as specified by DIMENSION statement in calling program

3.2.5.4 Outputs

C matrix (M, K) which will be the product of A matrix (M, N) and B matrix (N, K).

3.2.5.5 Storage Requirements

To be determined.

3.2.5.6 Description

The product of the (M, N) matrix A and (N, K) matrix B is a (M, K) matrix C whose elements are defined as:

$$c_{ij} = \sum_{L=1}^{N} a_{iL} \times b_{Lj}$$

3.2.5.7 Flow Chart

Not applicable.

3.2.5.8 Listing

See Appendix A

3.2.6 SOFTWARE COMPONENT NO. 6 (SUBMX)

SUBMX subtracts two matrices of type REAL.

3.2.6.1 Linkages

SUBMX is referenced by LMAOV and PSINV. SUBMX does not call any other subprograms.

3.2.6.2 Interfaces

Interface between SUBMX and LMAOV and PSINV is only via the calling arguments of SUBMX.

3.2.6.3 Inputs

Calling Sequence:

CALL SUBMX (A, B, C, M, N, MX)

Parameter	Dimension	In/Out	Definition
A	(M, N)	In	Two dimensional subtrahend matrix
B	(M, N)	In	Two dimensional minuend matrix
C	(M, N)	Out	Two dimensional matrix containing difference of A and B
M	1	In	Row dimension of A, B, and C
N	1	In	Column dimension of A, B, and C
MX	1	In	First dimension of A, B, and C as specified in DIMENSION statement of the calling program

3.2.6.4 Outputs

Matrix C (M, N) will contain the difference of matrix A (M, N) and matrix B (M, N).

3.2.6.5 Storage Requirements

To be determined.

3.2.6.6 Description

SUBMX consists of two nested Do-loops where I = 1, N and J = 1, M. The difference of the elements of the two matrices where

$C = c_{ij}$, $A = a_{ij}$ and $B = b_{ij}$ is:

$$c_{ij} = a_{ij} - b_{ij}$$

3.2.6.7 <u>Flow Chart</u>

Not applicable.

3.2.6.8 <u>Listing</u>

See Appendix A

4. OPERATION

4.1 USER DOCUMENTATION

Run Deck Set up:

```
                                                                 Col 61
@ Run   Badge ID, Div CODE, BOX #., Proj#, Proj#, C, Time, Page  NAME
@ SCH   7T = 1
@ ASG   A = X01189 or X01180
@ XQT   CUR
  TRW   A
  IN    A
@ XQT   GLMAOV

   [Input data as                                          ]
   [described in Section 3.2.1.3 and Section 3.2.2.3       ]

@ FIN
```

APPENDIX A

PROGRAM LISTINGS

```
@ FOR  SUBMX,SUBMX                                                                                        25 MAR 78            10:31:28. 93
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E1201CC1CA)
THIS COMPILATION WAS DONE ON 25 MAR 78 AT 10:31:28

   SUBROUTINE SUBMX       ENTRY POINT 000076

   STORAGE USED: CODE(1) 000111; DATA(0) 000033; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000037 105G       0001    000040 110G       0003 I 000001 I          0000   000003 INJP$      0000 I 000000 J


00100        1*       C*
00100        2*       C*       SUBROUTINE SUBMX DOES REAL MATRIX SUBTRACTION
00100        3*       C*
00101        4*                SUBROUTINE SUBMX(A,B,C,M,N,MX)
00103        5*                DIMENSION A(MX,N), B(MX,N), C(MX,N)
00104        6*                DO 10 J=1,N
00107        7*                DO 10 I=1,M
00112        8*                C(I,J)=A(I,J)- B(I,J)
00113        9*          10    CONTINUE
00116       10*                RETURN
00117       11*                END

        END OF COMPILATION:         NO DIAGNOSTICS.
```

```
@ FOR MULTMX,MULTMX                                                                             25 MAR 78          10:31:29.295
UNIVAC 1106 FORTRAN V EXEC II LEVEL 2CA  (EXEC8 LEVEL E1201D010A)
THIS COMPILATION WAS DONE ON 25 MAR 78 AT 10:31:29

   SUBROUTINE MULTMX    ENTRY POINT 000132

   STORAGE USED: CODE(1) 000156; DATA(0) 000047; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000050 106G       0001   000055 111G       0001   000057 115G       0000 I 000003 I          0000   000006 INJP$
    0000 I 000002 J          0000 I 000004 K          0000 D 000000 W


00100      1*      C*
00100      2*      C*      MULTMX SUBROUTINE DOES MATRIX MULTIPLICATION IN DOUBLE PRECISION
00100      3*      C*       BUT RESTORES RESULT AS SINGLE PRECISION TRUNCATED.
00100      4*      C*
00101      5*              SUBROUTINE MULTMX(A,B,C,M,L,N,MX,LX)
00103      6*              DIMENSION A(MX,L), B(LX,N), C(MX,N)
00104      7*              DOUBLE PRECISION W
00105      8*              DO 10 J=1,N
00110      9*              DO 10 I=1,M
00113     10*              C(I,J)=0.
00114     11*              DO 10 K=1,L
00117     12*              W = C(I,J)+ A(I,K)* B(K,J)
00120     13*              C(I,J) = W
00121     14*         10   CONTINUE
00125     15*              RETURN
00126     16*              END

34     END OF COMPILATION:        NO DIAGNOSTICS.
```

```
@FOR  TRANSP,TRANSP
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E12010010A)                                  25 MAR 78            10:31:30.362
THIS COMPILATION WAS DONE ON 25 MAR 78 AT 10:31:30

   SUBROUTINE TRANSP     ENTRY POINT 000074

   STORAGE USED: CODE(1) 000106; DATA(0) 000032; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000042 105G       0001    000044 110G       0003 I 000000 I          0000    000003 INJP$      0000 I 000001 J


00100       1*      C*
00100       2*      C*      TRANSP DOES REAL MATRIX TRANSPOSITION
00100       3*      C*
00101       4*              SUBROUTINE TRANSP(A,B,M,N,MX,NX)
00103       5*              DIMENSION A(MX,N) , B(NX,M)
00104       6*              DO 10 I=1,M
00107       7*              DO 10 J=1,N
00112       8*              B(J,I)=A(I,J)
00113       9*       10     CONTINUE
00116      10*              RETURN
00117      11*              END

    END OF COMPILATION:        NO DIAGNOSTICS.
```

```
@ FOR  PSINV,PSINV                                                                  25 MAR 78            10:31:31.430
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E1201C010A)
THIS COMPILATION WAS DONE ON 25 MAR 78 AT 10:31:31

    SUBROUTINE PSINV      ENTRY POINT 000432 .

    STORAGE USED: CODE(1) 000579; DATA(0) 000043; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003    TRANSP
     0004    MULTMX
     0005    SUBMX
     0006    NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001    000015 106G        0001    000044 117G      0001    000056 126G       0001    000117 137G       0001    000132 146G
     0001    000154 152G        0001    000221 164G      0001    000305 201G       0001    000346 213G       0001    000411 224G
     0001    000047 30L         0001    000123 60L       0001    000312 90L        0001    000352 95L        0000 R 000002 CC
     0000 R  000007 DN          0000 I  000001 I         0000    000013 INJP$      0000 I  000004 K          0000 I 000006 M
     0000 I  000005 N           0000 R  000003 SCALAR    0000 R  000000 XX


 00100      C*     SUBROUTINE PSINV COMPUTES THE MOORE-PENROSE PSEUDO-INVERSE OF A
 00100      C*      MATRIX(DENOTED A+) USING ALGORITHM KNOWN AS GREVILLE'S METHOD.
 00100      C*      IF 'A' MATRIX IS SQUARE AND FULL RANK THEN 'A' HAS AN INVERSE
 00100      C*           A+ = A(-1)
 00100      C*      ALSO IF 'B' IS A NONZERO PX1 VECTOR THEN
 00100      C*                B+ = (B(T)*B )(-1)*B(T)
 00100      C*
 00100      C*                                    B(T) BEING TRANSPOSE OF B
 00100      C*                                      AND (-1) DENOTING INVERSE.
 00100      C*
 00101              SUBROUTINE PSINV(AMX,OUTMX,NROW,NCOL,TOLENC,FCOL,TCOL,DK,CK,AK,
 00101             *DTR,BK,SCRACH)
 00103              DIMENSION AMX(NROW,NCOL),OUTMX(NCOL,NROW),FCOL(NROW),TCOL(NROW),
 00103             *DK(NCOL),CK(NROW),AK(NROW),DTR(NCOL),BK(NROW),SCRACH(NCOL,NROW)
 00103      C*
 00103      C*     COMPUTE PSEUDO INVERSE
 00104              XX=0.
 00105              DO 10 I=1,NROW
 00110              XX=AMX(I,1)**2 + XX
 00111        10    CONTINUE
 00113              CC = (TOLENC**2) * AMAX1(1.,XX)
 00114              IF(ABS(XX).GT.CC) GO TO 30
 00114      C*     IF FIRST COL VECTOR IS ZERO THEN FIRST ROW OF OUTMX IS ZERO
 00116              DO 20 I=1,NROW
 00121              OUTMX(1,I)=0.
 00122        20    CONTINUE
 00124              GO TO 60
 00124      C*     IF FIRST COL. IS NOT ZERO, THEN DO FIRST ROW OF AT
```

```
00125      29*      30    DO 4C I=1,NROW
00130      30*            FCOL(I) = AMX(I,1)
00131      31*      40    CONTINUE
00131      32*     C*     TRANSPOSE FIRST COLUMN
00133      33*            CALL TRANSP(FCOL,TCOL,NROW,1,NROW,1)
00133      34*     C*     MULT TRANSPOSE BY ORIGINAL COL., PRODUCT IS SCALAR
00134      35*            CALL MULTMX(TCOL,FCOL,SCALAR,1,NROW,1,1,NROW)
00134      36*     C*     MULT SCALAR BY TRANSPOSE
00135      37*            SCALAR = 1./SCALAR
00136      38*            DO 50 I=1,NROW
00141      39*            OUTMX(I,1)=FCOL(I) * SCALAR
00142      40*      50    CONTINUE
00144      41*      60    CONTINUE
00145      42*            DO 100 K=2,NCOL
00150      43*            N=K-1
00151      44*            DO 70 M=1,NROW
00154      45*            AK(M) = AMX(M,K)
00155      46*      70    CONTINUE
00157      47*            CALL MULTMX(OUTMX,AK,DK,N,NROW,1,NCOL,NROW)
00160      48*            CALL MULTMX(AMX ,DK,FCOL,NROW,N,1,NROW,NCOL)
00161      49*            CALL SUBMX(AK,FCOL,CK,NROW,1,NROW)
00161      50*     C*
00161      51*     C*     DETERMINE IF CK=0
00162      52*            XX=0.
00163      53*            DO 80 M=1,NROW
00166      54*            XX=CK(M)**2 + XX
00167      55*      80    CONTINUE
00171      56*            CC=(TOLERC**2) *AMAX1(1.,XX)
00172      57*            IF(ABS(XX).GT.CC) GO TO 90
00172      58*     C*
00172      59*     C*     IF CK = 0  THEN COMPUTE BK
00174      60*            CALL TRANSP(DK,DTR,N,1,NCOL,1)
00175      61*            CALL MULTMX(DTR,OUTMX,BK,1,N,NROW,1,NCOL)
00176      62*            CALL MULTMX(DTR,DK,SCALAR,1,N,1,1,NCOL)
00177      63*            DN=1. + SCALAR
00200      64*            DO 85 M=1,NROW
00203      65*            BK(M) = BK(M)/DN
00204      66*      85    CONTINUE
00206      67*            GO TO 95
00206      68*     C*
00206      69*     C*     IF CK.NE.0 ,BK=CK+
00206      70*     C*
00207      71*      90    CALL TRANSP(CK,TCOL,NROW,1,NROW,1)
00210      72*            CALL MULTMX(TCOL,CK,SCALAR,1,NROW,1,1,NROW)
00211      73*            SCALAR = 1./SCALAR
00212      74*            DO 92 M=1,NROW
00215      75*            BK(M) = TCOL(M) * SCALAR
00216      76*      92    CONTINUE
00220      77*      95    CONTINUE
00221      78*            CALL MULTMX(DK,BK,SCRACH,N,1,NROW,NCOL,1)
00222      79*            CALL SUBMX(OUTMX,SCRACH,OUTMX,N,NROW,NCOL)
00223      80*            DO 96 M=1,NROW
00226      81*            OUTMX(K,M) = BK(M)
00227      82*      96    CONTINUE
00231      83*     100    CONTINUE
00233      84*            RETURN
00234      85*            END
```

```
END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
@ FOR  LMADV,LMADV                                                                     25 MAR 78              10:31:33.718
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  (EXEC8 LEVEL E1201C010A)
THIS COMPILATION WAS DONE ON 25 MAR 78 AT 10:31:33

   SUBROUTINE LMADV      ENTRY POINT 004715

   STORAGE USED: CODE(1) 007252; DATA(0) 000776; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003    PSINV
    0004    MULTMX
    0005    SUBMX
    0006    TRANSP
    0007    NWDU$
    0010    NI02$
    0011    NRDU$
    0012    NI01$
    0013    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000    000103 1000F        0000    000135 1001F        0000    000073 1002F        0000    000075 1003F        0000    000113 1004F
    0000    000116 1005F        0000    000124 1007F        0001    003076 1025G        0001    003261 1052G        0001    003411 1075G
    0001    003434 1113G        0001    003531 1140G        0001    003534 1143G        0001    003674 1211G        0001    003714 1224G
    0001    004003 1236G        0001    004226 1277G        0001    000321 13L          0001    004223 1310G        0001    004303 1326G
    0001    004336 1346G        0001    004511 1375G        0001    000135 140G         0001    004540 1434G        0001    004554 1415G
    0001    000146 144G         0001    004635 1444G        0001    004652 1454G        0001    004732 1472G        0001    000591 15L
    0001    001615 150L         0001    000221 153G         0001    005171 1530G        0001    005217 1537G        0001    005233 1550G
    0001    001632 160L         0001    005444 1612G        0001    005455 1622G        0001    005535 1640G        0001    000723 17L
    0001    001647 170L         0001    001715 172L         0001    001723 173L         0001    000243 175G         0001    002167 174L
    0001    002167 180L         0000    000326 1999F        0001    002454 200L         0000    000163 2000F        0000    000222 2001F
    0000    000251 2002F        0000    000167 2003F        0000    000325 2004F        0001    002525 202L         0001    002530 203L
    0001    000257 206G         0001    000312 215G         0001    000401 235G         0001    000405 241G         0000    000107 2499F
    0001    000440 250G         0001    003544 250L         0000    000133 2509F        0001    003547 251L         0001    000463 265G
    0001    000477 275G         0000    000361 2999F        0001    000324 30L          0000    000370 3000F        0000    000366 3001F
    0000    000371 3002F        0000    000414 3003F        0000    000422 3004F        0000    000431 3005F        0000    000490 3006F
    0000    000443 3007F        0000    000461 3006F        0000    000471 3009F        0000    000477 3010F        0000    000456 3011F
    0000    000512 3012F        0000    000514 3015F        0000    000524 3018F        0000    000531 3020F        0000    000537 3021F
    0000    000543 3025F        0000    000564 3030F        0000    000577 3035F        0001    000532 364G         0000    000601 3040F
    0000    000617 3050F        0000    000631 3060F        0000    000633 3080F        0000    000647 3090F        0000    000667 3095F
    0001    000563 322G         0001    000567 326G         0001    004127 330L         0001    000622 335G         0001    004144 340L
    0001    003645 352G         0001    000661 362G         0001    000714 371G         0001    004564 410L         0001    004601 420L
    0001    001106 427G         0001    001111 432G         0001    001152 451G         0001    001200 463G         0001    005367 510L
    0001    005404 520L         0001    001424 521G         0001    001475 541G         0001    001702 610G         0001    001705 613G
    0001    002114 650G         0001    002114 653G         0001    000762 71L          0001    000776 72L          0001    002375 725G
    0001    001121 73L          0001    001124 74L          0001    002512 752G         0001    002515 755G         0001    001225 79L
    0000 R  000000 A            0000 C  000035 CC           0000 K  000041 CX           0000 K  000052 F            0000 R  000044 GARK
    0000 I  000004 HEQ          0000 R  000043 HS           0000 R  000042 HX           0000 I  000013 I            0000 I  000062 II
    0000 I  000022 IIKK         0000 I  000061 IJ           0000 I  000021 IK           0000 I  000023 IKK          0000    000702 INJP$
    0000 I  000010 IZERO        0000 I  000015 J            0000 I  000063 JJ           0000 I  000014 K            0000 I  000020 KI
    0000 I  000017 KK           0000 I  000016 K8           0000 I  000007 LIMIT        0000 I  000001 LP           0000 I  000024 LP1
    0000 I  000025 LP2          0000 I  000026 LP3          0000 I  000027 LP4          0000 I  000030 LP5          0000 I  000031 LP6
    0000 I  000032 LP7          0000 I  000033 LP8          0000 I  000065 M            0000 K  000006 NPY          0000 I  000050 NR
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0000 I 000011 NREAD      0000 I 000051 NRR        0000 I 000060 NT1        0000 I 000056 NT2        0000 I 000057 NT3
0000 I 000055 NT4        0000 I 000012 NWRITE     0000 R 000046 Q          0000 R 000047 Q1         0000 R 000067 Q2
0000 R 000066 R          0000 R 000036 RWRK       0000 I 000002 R1         0000 I 000003 R2         0000 R 000053 S1
0000 R 000059 S2         0000 I 000025 TEW        0000 R 000045 TRTOT      0000 R 000037 WW         0000 R 000040 WX
0000 R 000034 XX         0000 R 000064 YMEAN


00100      1*      C*     LHAOV IS TO COMPLETE AN ANALYSIS OF VARIANCE FOR UNBALANCED
00100      2*      C*      DESIGNS,DESIGNS WITH MISSING DATA, AND DESIGNS WITH MULTIPLE
00100      3*      C*      SOLUTIONS NORMAL EQUATIONS. ANALYSES INCLUDES ANALYSIS OF
00100      4*      C*      COVARIANCE AND ANALYSIS OF REGRESSION.
00100      5*      C*
00100      6*      C*      INPUTS - YVEC TO WORST ARE WORK ARRAYS WHOSE BASE ADDRESSES ARE
00100      7*      C*               COMPUTED IN CALLING PROG. STORAGE IS TAKEN FROM EXISTING
00100      8*      C*               STORAGE ALLOCATED IN CALLING PROG. NP,NQ,NM,NN ARE
00100      9*      C*               DIMENSIONS TO MATH MODELS AND NALYS IS NO. ANALYSES.
00100     10*      C*      OUTPUTS- ANALYSIS OF VARIANCE TABLE AND BEST LINEAR ESTIMATE(BLE)
00100     11*      C*               UNRESTRICTED,THE BLE RESTRICTED ONLY BY MODEL
00100     12*      C*               RESTRICTIONS, AND BLE RESTRICTED BY THE HYPOTHESIS.
00100     13*      C*
00101     14*             SUBROUTINE LHAOV(YVEC,XMX,BVEC,RMX,CTVEC,RINV,TVEC,GMX,GINV,HINV,
00101     15*            *CHV,HVEC,WKR,XTRP,HMX,GWKR,CHVEC,RWKR,CMX,CHMX,WKRS,CINV,MMX,ZWRK,
00101     16*            *ZWST,PWRK,MINV,PWST,ZTRP,PPWRK,EWRK,WARR,
00101     17*            *WORKST,NP,NQ,NM,NN,NALYS,TOLENC,NWS)
00103     18*             DATA A/'------'/
00105     19*             DATA LP/'('/
00107     20*             INTEGER R1,R2 ,NEW , TEW
00110     21*             REAL MMX
00111     22*             REAL NPY
00112     23*             DATA LIMIT/4200/
00114     24*             DIMENSION YVEC(NP),XMX(NP,NQ),BVEC(NQ),RMX(NM,NQ),CHV(NQ),
00114     25*            *RINV(NQ,NM),TVEC(NM),GMX(NM,NQ),GINV(NQ,NN),HVEC(NN) ,WKR(NQ,NQ),
00114     26*            *XTRP(NQ,NP),HMX(NN,NQ),GWKR(NN,NN),CHVEC(NN),RWKR(NM,NM),CTVEC(NM)
00114     27*            S,CHMX(NN,NQ),WKRS(NQ,NQ),MMX(NP,NQ),CMX(NQ,NQ),ZWRK(NP),ZWST(NP),
00114     28*            *PWRK(NQ,NP),PWST(NP,NP),HINV(NQ,NN),MINV(NQ,NP),CINV(NQ,NQ),
00114     29*            *ZTRP(NP),PPWRK(NP,NP)   ,WORKST(1) ,EWRK(NQ,NQ),WARR(NN,NQ)
00115     30*             NEW = 0
00116     31*             TEW = 0
00117     32*             IZERO = 0
00117     33*      C*
00117     34*      C*     READ IN YVEC AND XMX AT FIRST ANALYSIS
00120     35*             WRITE(6,1002)
00122     36*        1002 FORMAT(1H1,///5X,'INPUT DATA')
00123     37*             NREAD=NQ/8
00124     38*             IF(MOD(NQ,8).GT.0)NREAD=NREAD+1
00126     39*             NWRITE = NQ -7
00127     40*             NWRITE = NWRITE/8
00130     41*             IF(MOD(NWRITE,8).GT.0)NWRITE = NWRITE + 1
00132     42*             IF(NALYS.GT.1)GO TO 33
00134     43*             WRITE(6,1003)
00136     44*        1003 FORMAT(//5X,'Y VEC OR AND X MATRIX')
00137     45*             DO 20 I=1,NP
00142     46*             K=1
00143     47*             DO 10 J=1,NREAD
00146     48*             K8 = K + 7
00147     49*             IF(K8.GT.NQ)K8=NQ
```

ORIGINAL PAGE IS OF POOR QUALITY

```
00151           READ(5,1000)(XMX(I,KK),KK=K,K8)
00157      1000 FORMAT(8F10.6)
00160           K=K8
00161      10   CONTINUE
00163           READ(5,1001)YVEC(I)
00166      1001 FORMAT(F10.6)
00167           KI = 7
00170           IF(NQ.LE.KI)KI=NQ
00172           WRITE(6,2499)YVEC(I),(XMX(I,KK),KK=1,KI)
00201      2499 FORMAT(/15X,F10.6,7(3X,F10.6))
00202           IF(NQ.LE.7)GO TO 13
00204           IK = 8
00205           DO 12 J=1,NWRITE
00210           IIKK = IK + 7
00211           IF(IIKK.GT.NQ)IIKK = NQ
00213           WRITE(6,1004)(XMX(I,IKK),IKK=IK,IIKK)
00221           IK = IIKK
00222      12   CONTINUE
00224      1004 FORMAT(/15X,8(F10.6,3X))
00225      13   CONTINUE
00226      20   CONTINUE
00230      30   CONTINUE
00231           WRITE(6,1005)
00233      1005 FORMAT(//5X,'T VECTOR AND R MATRIX')
00234           DO 50 I=1,NH
00237           K=1
00240           DO 40 J=1,NREAD
00243           K8 = K + 7
00244           IF(K8.GT.NQ)K8=NQ
00246           READ(5,1000)(RMX(I,KK),KK=K,K8)
00254           K=K8
00255      40   CONTINUE
00257           READ(5,1001)TVEC(I)
00262           WRITE(6,2499)TVEC(I),(RMX(I,KK),KK=1,KI)
00271           IF(NQ.LE.7)GO TO 15
00273           IK = 8
00274           DO 14 J=1,NWRITE
00277           IIKK = IK + 7
00300           IF(IIKK .GT.NQ)IIKK = NQ
00302           WRITE(6,1004)(RMX(I,IKK),IKK=IK,IIKK)
00310           IK = IIKK
00311      14   CONTINUE
00313      15   CONTINUE
00314      50   CONTINUE
00316           WRITE(6,1007)
00320      1007 FORMAT(//5X,'H VECTOR AND LAMBDA MATRIX')
00321           DO 70 I=1,NH
00324           K=1
00325           DO 60 J=1,NREAD
00330           K8 = K + 7
00331           IF(K8.GT.NQ)K8=NQ
00333           READ(5,1000)(GMX(I,KK),KK=K,K8)
00341           K=K8
00342      60   CONTINUE
00344           READ(5,1001)HVEC(I)
00347           WRITE(6,2499)HVEC(I),(GMX(I,KK),KK=1,KI)
00356           IF (NQ.LE.7)GO TO 17
00360           IK = 8
```

```
00361              DO 16 J = 1,NWRITE
00364              IIKK = IK + 7
00365              IF(IIKK.GT.NQ)IIKK = NQ
00367              WRITE(6,1004)(GMX(I,IKK),IKK=IK,IIKK)
00375              IK = IIKK
00376        16    CONTINUE
00400        17    CONTINUE
00401        70    CONTINUE
00401        C*
00401        C*    COMPUTE  W=GMX(I-(RMX+RMX)  I=IDENTITY MATRIX
00401        C*
00401        C*    COMPUTE BASE ADDRESSES FOR SUBROUTINE PSINV
00401        C*
00403              LP1=NM+1
00404              LP2=NM+1+LP1
00405              LP3=NQ+1+LP2
00406              LP4=NM+1+LP3
00407              LP5=NM+1+LP4
00410              LP6=NQ+1+LP5
00411              LP7=NM+1+LP6
00412              LP8=NM*NQ+1+LP7
00413              IF(LP8.LE.LIMIT)GO TO 72
00415        71    LP7 = LIMIT - LP8
00416              WRITE(6,2509)LP7
00421        2509  FORMAT(5X'***DIMENSIONS OF MATH MODELS EXCEEDS STORAGE CAPACITY O
00421             *F LMADV SUBROUTINE  ** ANALYSIS ASKS FOR',I5,' MORE LOCATIONS THAN
00421             *AVAILABLE')
00422              RETURN
00423        72    CONTINUE
00424              CALL PSINV(RMX,RINV,NM,NQ,TOLENC,WORKST(1),WORKST(LP1),WORKST(LP2)
00424             *,WORKST(LP3),WORKST(LP4),WORKST(LP5),WORKST(LP6),WORKST(LP7))
00425              CALL MULTMX(RINV,RMX,WKR,NQ,NM,NQ,NQ,NM)
00425        C*
00426              DO 78 J=1,NQ
00431              DO 75 I=1,NQ
00434              IF(J.EQ.I)GO TO 73
00436              WKR(J,I) = 0. - WKR(J,I)
00437              GO TO 74
00440        73    WKR(J,I) = 1. - WKR(J,I)
00441        74    CONTINUE
00442        75    CONTINUE
00444        78    CONTINUE
00444        C*
00446              CALL MULTMX(GMX,WKR,HMX,NM,NQ,NQ,NM,NQ)
00446        C
00446        C*    DETERMINE IF TVEC = 0 AND HVEC = 0
00446        C
00447              XX = 0.
00450              DO 76 I=1,NM
00453              XX=XX+HVEC(I)**2
00454        76    CONTINUE
00456              CC = (TOLENC**2) * AMAX1(1.,XX)
00457              IF(ABS(XX).GT.CC) GO TO 79
00461              XX = 0.
00462              DO 77 I=1,NM
00465              XX = XX + TVEC(I)**2
00466        77    CONTINUE
00470              CC = (TOLENC**2) * AMAX1(1.,XX)
```

42

```
00471     166*              IF(ABS(XX).GT.CC) GO TO 79
00473     167*              IZERO = I
00474     168*              GO TO 180
00475     169*        79    CONTINUE
00475     170*        C*
00475     171*        C*    COMPUTE GMX(GMX+)*HMX=CHVEC
00476     172*              LP1=NN+1
00477     173*              LP2=NN+1+LP1
00500     174*              LP3=NN+1+LP2
00501     175*              LP4=NN+1+LP3
00502     176*              LP5=NN+1+LP4
00503     177*              LP6=NQ+1+LP5
00504     178*              LP7=NN+1+LP6
00505     179*              LP8=NN*NN+1+LP7
00506     180*              IF(LP8.GT.LIMIT) GO TO 71
00510     181*              CALL PSINV(GMX,GINV,NN,NQ,TOLENC,WORKST(1),WORKST(LP1),WORKST(LP2),
00510     182*             *1WORKST(LP3),WORKST(LP4),WORKST(LP5),WORKST(LP6),WORKST(LP7))
00510     183*        C*
00511     184*              CALL MULTMX(GMX,GINV,GMXR,NN,NN,NN,NN,NQ)
00512     185*              CALL MULTMX(GMXR,HVEC,CHVEC,NN,NN,1,NN,NN)
00512     186*        C*
00512     187*        C*    CALCULATE HMX(HINV)*TVEC=CTVEC
00512     188*        C*
00513     189*              CALL MULTMX(HMX,HINV,HMXR,NN,NN,NN,NN,NQ)
00514     190*              CALL MULTMX(HMXR,TVEC,CTVEC,NN,NN,1,NN,NN)
00514     191*        C*    COMPARE CHVEC TO HVEC
00515     192*              XX=0.
00516     193*              HH=0.
00517     194*              HX = 0.
00520     195*              DO 80 I=1,NN
00523     196*              XX=CHVEC(I)**2 + XX
00524     197*              HH=HVEC(I)**2 + HH
00525     198*              HX=CHVEC(I)*HVEC(I)  + HX
00526     199*        80    CONTINUE
00530     200*              CX=XX-2*HX+HH
00531     201*              HX=XX+HH
00532     202*              HH= TOLENC**2 * AMAX1(1.,HX)
00533     203*              IF(ABS(CX).LT.HH)HEQ = 1
00533     204*        C*
00533     205*        C*    COMPARE CTVEC TO TVEC
00535     206*              XX=0.
00536     207*              HH=0.
00537     208*              HX = 0.
00540     209*              DO 90 I=1,NN
00543     210*              XX=CTVEC(I)**2+XX
00544     211*              HH=TVEC(I)**2 +HH
00545     212*              HX=CTVEC(I)*TVEC(I)  + HX
00546     213*        90    CONTINUE
00550     214*              CX=XX-2*HX+HH
00551     215*              HX=XX+HH
00552     216*              HH = TOLENC**2 * AMAX1(1.,HX)
00553     217*              IF(ABS(CX).LT.HH)TEQ = 1
00553     218*        C*
00555     219*              IF(HEQ.EQ.1.AND.TEQ.EQ.1)GO TO 170
00557     220*              IF(HEQ.EQ.1.AND.TEQ.EQ.0)GO TO 150
00561     221*              IF(HEQ.EQ.0.AND.TEQ.EQ.1)GO TO 160
00561     222*        C*
00561     223*        C*     HEQ=FALSE AND TEQ=FALSE
```

```
00563     224*          WRITE(6,2000)
00565     225*     2000 FORMAT(////,5X,'*********')
00566     226*          WRITE(6,2003)
00570     227*     2003 FORMAT(//,5X,'REJECT H(0) WITH PROB(TYPE I ERROR)=PROB(TYPE II ERR
00570     228*         *OR)=0.',//,5X,'NEITHER THE ASSUMED RESTRICTIONS NOR THE HYPOTHESIS
00570     229*         * IS MATHEMATICALLY CONSISTENT.')
00571     230*          RETURN
00572     231*      150 WRITE(6,2000)
00574     232*          WRITE(6,2001)
00576     233*     2001 FORMAT(//,5X,'H(0) IS A LEGITIMATE HYPOTHESIS BUT THE MODELL IS NOT
00576     234*         * MATHEMATICALLY CONSISTENT.',//,5X,'RECONSIDER THE ASSUMED RESTRIC
00576     235*         *TIONS.')
00577     236*          RETURN
00600     237*      160 WRITE(6,2000)
00602     238*          WRITE(6,2002)
00604     239*     2002 FORMAT(//5X,'REJECT H(0) WITH PROB(TYPE I ERROR)=PROB(TYPE II ERRO
00604     240*         *R)=0.',//,5X,'THE ASSUMED RESTRICTIONS ARE MATHEMATICALLY CONSISTE
00604     241*         *NT. BUT THE HYPOTHESIS IS NOT CONSISTENT.')
00605     242*          RETURN
00605     243*     C*
00605     244*     C    IF HMX(XMX(I-RINV(RMX))+*(XMX(I-RINV(RMX)) .NE. HMX   PRINT
00605     245*     C     MESSAGE BELOW
00606     246*      170 CALL MULTMX(RINV,RMX,WKR,NQ,NM,NQ,NQ,NM)
00607     247*          DO 175 J=1,NQ
00612     248*          DO 174 I=1,NQ
00615     249*          IF(J.EQ.I)GO TO 172
00617     250*          WKR(J,I) = 0. - WKR(J,I)
00620     251*          GO TO 173
00621     252*      172 WKR(J,I) = 1.-WKR(J,I)
00622     253*      173 CONTINUE
00623     254*      174 CONTINUE
00625     255*      175 CONTINUE
00627     256*          CALL MULTMX(XMX,WKR,HMX,NP,NQ,NQ,NP,NQ)
00630     257*          LP1 = NP + 1
00631     258*          LP2 = NP + 1 +LP1
00632     259*          LP3 = NQ + 1 + LP2
00633     260*          LP5 = NP + 1 + LP4
00634     261*          LP6 = NQ + 1 + LP5
00635     262*          LP7 = NP + 1 + LP6
00636     263*          LP8 = NP * NQ + 1 + LP7
00637     264*          IF(LP8.GT.LIMIT) GO TO 71
00641     265*          CALL PSINV(HMX,HINV,NP,NQ,TOLENC,WORKST(1),WORKST(LP1),WORKST(LP2)
00641     266*         *,WORKST(LP3),WORKST(LP4),WORKST(LP5),WORKST(LP6),WORKST(LP7))
00642     267*          CALL MULTMX(HINV,HMX,EWRK,NQ,NP,NQ,NQ,NP)
00643     268*          CALL MULTMX(HMX,EWRK,WARR,NN,NQ,NQ,NN,NQ)
00643     269*     C
00643     270*     C    COMPARE , IF NOT EQUAL, PRINT MESSAGE AND CONTINUE
00643     271*     C
00644     272*          XX = 0.
00645     273*          WW = 0.
00646     274*          WX = 0.
00647     275*          DO 177 I = 1,NQ
00652     276*          DO 176 J = 1,NN
00655     277*          XX = WARR(J,I)**2 + XX
00656     278*          WW = HMX(J,I)**2 + WW
00657     279*          WX = WARR(J,I) * HMX(J,I) + WX
00660     280*      176 CONTINUE
00662     281*      177 CONTINUE
```

```
00664    282.          CX = XX - (2. * WX) + WW
00665    283.          HX = XX + WW
00666    284.          HW = (TOLENC**2)*AMAX1(1.,HX)
00667    285.          IF(ABS(CX).LT.HW)GO TO 178
00671    286.          WRITE(6,2033)
00673    287.          WRITE(6,1999)
00675    288.     1999 FORMAT(//5X,'THE HYPOTHESIS FUNCTION.LAMBDA.BETA. IS NOT ESTIMABLE
00675    289.         *. ANALYSIS CONTINUES.')
00676    290.      178 CONTINUE
00676    291.    C.    CALCULATE THEN COMPARE(HMX)HMX(+).(HVEC-GMX(RMX+)TVEC)=(HVEC-GMX.
00676    292.    C.    (RMX+)TVEC
00676    293.    C.
00677    294.      180 CALL MULTMX(GMX,RINV,CHMX,NN,NQ,NN,NN,NQ)
00677    295.    C.
00677    296.    C.    CHVEC CONTAINS(HVEC-GMX(RINV)TVEC)
00700    297.          CALL MULTMX(CHMX,TVEC,CHVEC,NN,NM,1,NN,NM)
00701    298.          CALL SUBMX(HVEC,CHVEC,CHVEC,NN,1,NN)
00701    299.    C.
00702    300.          LP1=NN+1
00703    301.          LP2=NN+1+LP1
00704    302.          LP3=NQ+1+LP2
00705    303.          LP4=NN+1+LP3
00706    304.          LP5=NN+1+LP4
00707    305.          LP6=NQ+1+LP5
00710    306.          LP7=NN+1+LP6
00711    307.          LP8=NN*NQ+1+LP7
00712    308.          IF(LP8.GT.LIMIT)GO TO 71
00714    309.          CALL PSINV(HMX,HINV,NN,NQ,TOLENC,WORKST(1),WORKST(LP1),WORKST(LP2)
00714    310.         *,WORKST(LP3),WORKST(LP4),WORKST(LP5),WORKST(LP6),WORKST(LP7))
00715    311.          IF(IZERO.EQ.1)GO TO 200
00717    312.          CALL MULTMX(HMX,HINV,GWRK,NN,NQ,NN,NN,NQ)
00717    313.    C.    GWRK CONTAINS HH+
00717    314.    C.    HVEC CONTAINS HH+*CHVEC
00720    315.          CALL MULTMX(GWRK,CHVEC,HVEC,NN,NN,1,NN,NN)
00721    316.          XX=0.
00722    317.          WW=0.
00723    318.          WX = 0.
00724    319.          DO 190 I=1,NN
00727    320.          XX=CHVEC(I)**2+XX
00730    321.          WW=HVEC(I)**2+WW
00731    322.          WX=CHVEC(I)*HVEC(I) + WX
00732    323.      190 CONTINUE
00734    324.          CX=XX-2*WX+WW
00735    325.          HX=XX+WW
00736    326.          HW = TOLENC**2 * AMAX1(1.,HX)
00737    327.          IF(ABS(CX).LT.HW)GO TO 200
00741    328.          WRITE(6,2033)
00743    329.          WRITE(6,2034)
00745    330.     2004 FORMAT(//,5X,'REJECT H(0) WITH PROB(TYPE I ERROR)=PROB(TYPE II ERR
00745    331.         *OR)=0.',//5X,'THE RESTRICTIONS AND HYPOTHESES ARE SEPARATELY CONSI
00745    332.         *STENT, BUT NOT JOINTLY COMPATIBLE.')
00746    333.          RETURN
00746    334.    C.
00746    335.    C.    CALCULATE H=XHX(I-(RMX+)RMX-(HMX+)HMX)
00746    336.    C.    WKR CONTAINS(H+)H
00747    337.      200 CALL MULTMX(HINV,HMX,WKR,NQ,NN,NQ,NQ,NN)
00747    338.    C.
00747    339.    C.    WKRS CONTAINS (RMX+)RMX
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
00750            CALL MULTMX(RINV,RMX,WKRS,NQ,NM,NQ,NQ,NM)
00751            DO 205 J=1,NQ
00754            DO 204 I=1,NQ
00757            IF(J.EQ.I) GO TO 202
00761            WKRS(I,J) = 0. - WKRS(I,J)
00762            GO TO 203
00763      202   WKRS(I,J) = 1. - WKRS(I,J)
00764      203   CONTINUE
00765      204   CONTINUE
00767      205   CONTINUE
00771            CALL SUBMX(WKRS,WKR,WKR,NQ,NQ,NQ)
00772            CALL MULTMX(XMX,WKR,HMA,NP,NQ,NQ,NP,NQ)
00772     C*     M=MMX
00772     C*     CALCULATE C=(I-(RMX+)RMX)*TRP(XMX)(I-(RMX+)RMX)
00772     C*     WKRS CONTAINS I-(RMX+)RMX
00772     C*
00772     C*     TAKE TRANSPOSE OF XMX = XTRP
00773            CALL TRANSP(XMX,XTRP,NP,NQ,NP,NQ)
00773     C*     WRK CONTAINS XTRP*XMX
00774            CALL MULTMX(XTRP,XMX,WKR,NQ,NP,NQ,NQ,NP)
00775            LP4 = NP + 1 + LP3
00775     C*
00775     C*    EWKR = XTRP(XMX)(I-(RMX+)RMX)
00776            CALL MULTMX(WKR,WKRS,EWRK,NQ,NQ,NQ,NQ,NQ)
00776     C*
00776     C*     C=CMX
00777            CALL MULTMX(WKRS,EWRK,CMX,NQ,NQ,NQ,NQ,NQ)
00777     C*
00777     C*     CALCULATE Z=YVEC-XMX(RMX+)TVEC-XMX(HMX+)(HVEC-GMX(RMX+)TVEC)
00777     C*     CHVEC CONTAINS (HVEC-GMX(RMX+)TVEC)
00777     C*
00777     C*     CHV CONTAINS(HMX+)CHVEC
01000            CALL MULTMX(HINV,CHVEC,CHV,NQ,NN,1,NQ,NH)
01000     C*     ZWRK=XMX * CHV
01001            CALL MULTMX(XMX,CHV,ZWRK,NP,NQ,1,NP,NQ)
01001     C*     CHV NOW CONTAINS (RMX+)*TVEC
01002            CALL MULTMX(RINV,TVEC,CHV,NQ,NM,1,NQ,NM)
01002     C*     ZWST = XMX*CHV.
01003            CALL MULTMX(XMX,CHV,ZWST,NP,NQ,1,NP,NQ)
01004            CALL SUBMX(YVEC,ZWST,ZWST,NP,1,NP)
01005            CALL SUBMX(ZWST,ZWRK,ZWRK,NP,1,NP)
01005     C*     Z=ZWRK
01005     C*
01005     C*     CALCULATE NP-KI = TR(I-XMX(CMX+)XTRP . SOLVE FOR KI(ROUNDED INT.)
01006            LP1= NQ + 1
01007            LP2= NQ + 1 + LP1
01010            LP3 = NQ + 1 + LP2
01011            LP4 = NQ + 1 + LP3
01012            LP5 = NQ + 1 + LP4
01013            LP6 = NQ + 1 + LP5
01014            LP7 = NQ + 1 + LP6
01015            LP8 = NQ*NQ + 1 + LP7
01016            IF(LP8.GT.LIMIT)GO TO 71
01020            CALL PSINV(CMX,CINV,NQ,NQ,TOLENC,WORKST(1),WORKST(LP1)
01020           *,WORKST(LP2),WORKST(LP3),WORKST(LP4),WORKST(LP5),WORKST(LP6)
01020           *,WORKST(LP7))
01021            CALL MULTMX(CINV,XTRP,PWRK,NQ,NQ,NP,NQ,NQ)
01022            CALL MULTMX(XMX,PWRK,PWST,NP,NQ,NP,NP,NQ)
```

```
01022          C.
01022          C.     PWST CONTAINS (XMA(CINV)XTRP)
01022          C.     SUBTRACT IDENTITY MATRIX FROM PWST AND TAKE TRACE
01023                 TRTOT=0.
01024                 DO 215 I=1,NP
01027                 TRTOT  = TRTOT + (1. - PWST(I,I))
01030           215   CONTINUE
01032                 R1=FLOAT(NP)- TRTOT + .5
01032          C.
01032          C.     COMPUTE R1-R2=TR(XMX(CINV)XTRP-MMX(MINV)) SOLVE FOR R2(ROUND INT)
01032          C.
01033                 LP1=NP+1
01034                 LP2=NP+1+LP1
01035                 LP3=NQ+1+LP2
01036                 LP4=NP+1+LP3
01037                 LP5=NP+1+LP4
01040                 LP6=NQ+1+LP5
01041                 LP7=NP+1+LP6
01042                 LP8=NP*NQ+1 + LP7
01043                 IF(LP8.GT.LIMIT) GO TO 71
01045                 CALL PSINV(MMX,MINV,NP,NQ,TOLENC,WORKST(1),WORKST(LP1),WORKST(LP2)
01045                .,WORKST(LP3),WORKST(LP4),WORKST(LP5),WORKST(LP6),WORKST(LP7))
01046                 CALL MULTMX(MMX,MINV,PPWRK,NP,NQ,NP,NP,NQ)
01047                 CALL SUBMX(PWST,PPWRK,PPWRK,NP,NP,NP)
01050                 TRTOT=0.
01051                 DO 220 I=1,NP
01054                 TRTOT=TRTOT + PPWRK(I,I)
01055           220   CONTINUE
01057                 R2 = FLOAT(R1) - TRTOT + .5
01057          C.
01057          C.     CALCULATE Q3=ZTRP(XMA(CMX+)XTRP)ZMX
01057          C.     PWST CONTAINS XMA(CMA+)XTRP
01060                 CALL MULTMX(PWST,ZWRK,ZWST,NP,NP,1,NP,NP)
01061                 CALL TRANSP(ZWRK,ZTRP,NP,1,NP,1)
01062                 CALL MULTMX(ZTRP,ZWST,W,1,NP,1,1 ,NP)
01062          C
01062          C.     CALCULATE Q2=ZTRP*MMX(MMX+)ZMX
01062          C.     PPWRK CONTAINS MMX(MMX+)
01063                 CALL MULTMX(MMX,MINV,PPWRK,NP,NQ,NP,NP,NQ)
01063          C
01064                 CALL MULTMX(PPWRK,ZWRK,ZWST,NP,NP,1,NP,NP)
01065                 CALL MULTMX(ZTRP,ZWST,W1,1,NP,1,1,NP)
01065          C.     PRINT AOV TABLE
01066                 WRITE(6,2999)
01070          2999   FORMAT(1H1,///,5X,'AOV TABLE')
01071                 WRITE(6,3000)
01073                 WRITE(6,3001)(A,J=1,21)
01101          3001   FORMAT(/,1X,21A6)
01102          3000   FORMAT(////)
01103                 WRITE(6,3002)
01105          3002   FORMAT(T6,'SOURCE OF',T29,'DEGREE OF',T58,'SUM OF SQUARES',T83,
01105                .'MEAN SQUARE',T107,'VALUE OF F-STATISTIC')
01106                 WRITE(6,3003)
01110          3003   FORMAT(T6,'VARIATION',T30,'FREEDOM')
01111                 WRITE(6,3001)(A,J=1,21)
01117                 WRITE(6,3004)R1,Q
01123          3004   FORMAT(///,T6,'DUE TO B', T33, I3  ,T56,E15.9)
01124                 WRITE(6,3005)R2,W1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
01130     456.      3005 FORMAT(///,T6,'DUE TO B(N)',T33, I3  ,T56,E15.9)
01131     457.           WRITE(6,3006)
01133     458.      3006 FORMAT(T6,'(UNADJ.)')
01133     459.      C.
01133     460.      C.     COMPUTE    Q3=ZTRP(I-XMX(CMX+)XTRP)ZMX AND
01133     461.      C.                Q1=ZTRP(XMX(CMX+)XTRP - MMX(MMX+)ZMX
01133     462.      C.
01133     463.      C.
01133     464.      C.     XMX(CMX+)XTRP IS IN PWST AND MMX(MMX+) IS IN PPWRK
01134     465.           CALL SUBMX(PWST,PPWRK,PPWRK,NP,NP,NP)
01135     466.           CALL MULTMX(PPWRK,ZWRK,ZWST,NP,NP,1,NP,NP)
01136     467.           CALL MULTMX(ZTRP,ZWST,Q1,1,NP,1,1 ,NP)
01137     468.           DO 253 I=1,NP
01142     469.           DO 252 J=1,NP
01145     470.           IF(J.EQ.I)GO TO 250
01147     471.           PWST(J,I) = 0. - PWST(J,I)
01150     472.           GO TO 251
01151     473.       250 PWST(J,I) = 1. - PWST(J,I)
01152     474.       251 CONTINUE
01153     475.       252 CONTINUE
01155     476.       253 CONTINUE
01157     477.           CALL MULTMX(PWST,ZWRK,ZWST,NP,NP,1,NP,NP)
01160     478.           CALL MULTMX(ZTRP,ZWST,Q,1,NP,1,1 ,NP)
01161     479.           NR=NP -R1
01162     480.           NRR=R1-R2
01163     481.           F=Q1/Q * NR/NRR
01164     482.           S1=Q1/NRR
01165     483.           S2=Q/NR
01166     484.           WRITE(6,3007)NRR,Q1,S1,F
01174     485.      3007 FORMAT(///,T6,'DUE TO B(H)',T33, I3  ,T56,E15.9,T81,E15.9,T106,
01174     486.          .E15.9)
01175     487.           WRITE(6,3011)
01177     488.      3011 FORMAT(T6,'(ADJ)')
01200     489.           WRITE(6,3008)NR,Q,S2
01205     490.      3008 FORMAT(///,T6,'ERROR',T33, I3  ,T56,E15.9,T81,E15.9)
01205     491.      C.
01205     492.      C.     CALCULATE  Q=ZTRP(ZMX)
01206     493.           CALL MULTMX(ZTRP,ZWRK,Q,1,NP,1,1 ,NP)
01207     494.           WRITE(6,3021)(A,J=1,21)
01215     495.           WRITE(6,3009)NP,Q
01221     496.      3009 FORMAT(T6,'TOTAL',T33,I3,T56,E15.9)
01222     497.           WRITE(6,3021)(A,J=1,21)
01222     498.      C.
01222     499.      C.     PWRK CONTAINS(XMX+)
01230     500.           CALL PSINV(XMX,PWRK,NP,NQ,TOLENC,WORKST(1),WORKST(LP1),WORKST(LP2)
01230     501.          .,WORKST(LP3),WORKST(LP4),WORKST(LP5),WORKST(LP6),WORKST(LP7))
01231     502.           CALL MULTMX(PWRK,YVEC,BVEC,NQ,NP,1,NQ,NP)
01232     503.           WRITE(6,3013)
01234     504.      3013 FORMAT(1H1,///5X,'THE BEST LINEAR ESTIMATE (BLE) UNRESTRICTED IS-'
01234     505.          .,//)
01235     506.           DO 260 I=1,NQ
01240     507.           WRITE(6,3012)BVEC(I)
01243     508.      3012 FORMAT(T25,E15.9)
01244     509.       260 CONTINUE
01244     510.      C.
01244     511.      C      CALCULATE COVARIENCE MATRIX
01246     512.           CALL MULTMX(XTRP,XMX,WKR,NQ,NP,NQ,NQ,NP)
01247     513.           LP1= NQ + 1
```

```
      LP2= NQ + 1 + LP1
      LP3 = NQ + 1 + LP2
      LP4 = NQ + 1 + LP3
      LP5 = NQ + 1 + LP4
      LP6 = NQ + 1 + LP5
      LP7 = NQ + 1 + LP6
      LP8 = NQ*NQ + 1 + LP7
      CALL PSINV(WKR,WKRS,NQ,NQ,TOLENC,WORKST(1),WORKST(LP1),WORKST(LP2)
     *,WORKST(LP3),WORKST(LP4),WORKST(LP5),WORKST(LP6),WORKST(LP7))
      NT4=0
 330  NT2=0
      NT3=NT4+1
      NT4=NT3+19
      IF(NT4.GT.NQ)NT4=NQ
 340  NT1=NT2+1
      NT2=NT1+5
      IF(NT2.GT.NQ)NT2=NQ
      WRITE(6,3015)
 3015 FORMAT(////,5X,'BLE UNRESTRICTED COVARIENCE MATRIX')
      WRITE(6,3018)(LP,IJ,IJ=NT1,NT2)
 3018 FORMAT(//,T12,6 (A1 ,I2,')',15X))
      II = NT1
      JJ=NT1-1
      DO 350 I=NT3,NT4
      IF(II.NE.I)WRITE(6,3021)I
      IF(II.EQ.I)JJ = JJ + 1
      IF(JJ.GT.NT2)JJ=NT2
      IF(II.EQ.I)WRITE(6,3020)I,(WKRS(I,J),J=NT1,JJ)
 3020 FORMAT(/2X,'(',I2,')',6 (3X,E15.9))
 3021 FORMAT(/2X,'(',I2,')')
      IF(II.EQ.I)II=II + 1
 350  CONTINUE
      IF(NT2.LT.NQ)GO TO 340
      IF(NT4.LT.NQ)GO TO 330
C
C.    CALCULATE MEAN OF YVEC
      YMEAN = 0
      DO 351 M=1,NP
      YMEAN = YMEAN + YVEC(M)
 351  CONTINUE
      YMEAN = YMEAN / FLOAT(NP)
      NPY = FLOAT(NP) * (YMEAN**2)
C.
C.    CALCULATE SQUARED MULTIPLE CORRELATION COEFFICIENT
      CALL MULTMX(XTRP,YVEC,CHV,NQ,NP,1,NQ,NP)
      CALL MULTMX(BVEC,CHV,R,1,NQ,1,1,NQ)
      R = R - NPY
C.
      CALL TRANSP(YVEC,ZWST,NP,1,NP,1)
      CALL MULTMX(ZWST,YVEC,Q2,1,NP,1,1 ,NP)
      Q2 = Q2 - NPY
      R = R / Q2
      WRITE(6,3025)R
 3025 FORMAT(1H////,5X='BLE UNRESTRICTED SQUARED MULTIPLE CORRELATION CO
     *EFFICIENT',///,T32,E15.9)
      CALL MULTMX(RINV,TVEC,CHV,NQ,NM,1,NQ,NM)
      CALL MULTMX(XMX,CHV,ZWRK,NP,NQ,1,NP,NQ)
      CALL SUBMX(YVEC,ZWRK,ZWRK,NP,1,NP)
```

```
01373             CALL MULTMX(XTRP,ZWRK,BVEC,NQ,NP,1,NQ,NP)
01374             DO 380 I=1,NQ
01377             EWRK(I,1) = CHV(I)
01400       380   CONTINUE
01402             CALL MULTMX(CINV,BVEC,CHV ,NQ,NQ,1,NQ,NQ)
01402       C*
01403             DO 390 I=1,NQ
01406             BVEC(I) = EWRK(I,1) + CHV(I)
01407       390   CONTINUE
01407       C*
01411             WRITE(6,3030)
01413       3030  FORMAT(1H1,////5X,'THE BLE RESTRICTED ONLY BY THE MODEL RESTRICTIO
01413            *NS IS -',//)
01414             DO 400 I=1,NQ
01417             WRITE(6,3035)BVEC(I)
01422       3035  FORMAT(T25,E15.9)
01423       400   CONTINUE
01423       C*
01423       C*    COVARIENCE MATRIX FOR THE BLE RESTRICTED IS CMX+
01423       C*
01425             NT4=0
01426       410   NT2=0
01427             NT3=NT4+1
01430             NT4=NT3+19
01431             IF(NT4.GT.NQ)NT4=NQ
01433       420   NT1=NT2+1
01434             NT2=NT1+5
01435             IF(NT2.GT.NQ)NT2=NQ
01437             WRITE(6,3040)
01441       3040  FORMAT(////,5X,'COVARIENCE MATRIX OF BLE RESTRICTED ONLY BY THE MO
01441            *DEL RESTRICTIONS')
01442             WRITE(6,3018)(LP,IJ,IJ=NT1,NT2)
01451             II = NT1
01452             JJ=NT1-1
01453             DO 430 I=NT3,NT4
01456             IF(II.NE.1)WRITE(6,3021)I
01462             IF(II.EQ.1)JJ = JJ + 1
01464             IF(JJ.GT.NT2)JJ=NT2
01466             IF(II.EQ.1)WRITE(6,3020)I,(CINV(II,J),J=NT1,JJ)
01476             IF(II.EQ.1)II=II + 1
01500       430   CONTINUE
01502             IF(NT2.LT.NQ)GO TO 420
01504             IF(NT4.LT.NQ)GO TO 410
01504       C*
01504       C*    COMPUTE BLE RESTRICTED BY HYPOTHESIS
01504       C*
01504       C*    CHVEC CONTAINS HVEC-GMA(RMX+)TVEC
01506             CALL MULTMX(HINV,CHVEC,BVEC,NQ,NN,1,NQ,NN)
01507             CALL MULTMX(XMX,BVEC,ZWRK,NP,NQ,1,NP,NQ)
01507       C*   (RMX+)TVEC =CHV
01510             CALL MULTMX(RINV,TVEC,CHV,NQ,NM,1,NQ,NM)
01511             CALL MULTMX(XMX,CHV,ZWST,NP,NQ,1,NP,NQ)
01512             CALL SUBMX(TVEC,ZWST,ZWST,NP,1,NP)
01513             CALL SUBMX(ZWST,ZWRK,ZWST,NP,1,NP)
01514             LP1=NP+1
01515             LP2=NP+1+LP1
01516             LP3=NQ+1+LP2
01517             LP4=NP+1+LP3
```

```
01520          LP5=NP+1+LP4
01521          LP6=NQ+1+LP5
01522          LP7=NP+1+LP6
01523          LP8=NP*NQ+1 + LP7
01524          IF(LP8.GT.LIMIT) GO TO 71
01526          CALL PSINV(MMX,MINV,NP,NQ,TOLENC,WORKST(1),WORKST(LP1),WORKST(LP2)
01526         *,WORKST(LP3),WORKST(LP4),WORKST(LP5),WORKST(LP6),WORKST(LP7))
01527          DO 440 I=1,NQ
01532          CHV(I)=CHV(I) + BVEC(I)
01533    440   CONTINUE
01535          CALL MULTMX(MINV,ZAST,BVEC,NQ,NP,1,NQ,NP)
01536          DO 450 I=1,NQ
01541          CHV(I)=CHV(I) + BVEC(I)
01542    450   CONTINUE
01544          WRITE(6,3055)
01546   3055   FORMAT(1H1,7/5X,'THE BLE RESTRICTED BY THE HYPOTHESIS IS -',/)
01547          DO 460 I=1,NQ
01552          WRITE(6,3060)CHV(I)
01555   3060   FORMAT(T20,E15.9)
01556    460   CONTINUE
01556   C*
01556   C*     COVARIENCE MATRIX FOR BLE RESTRICTED BY HYPOTHESIS =
01556   C*     INVERSE OF(MMX TRANSPOSE * MMX)
01556   C*
01560          CALL TRANSP(MMX,MINV,NP,NQ,NP,NQ)
01561          CALL MULTMX(MINV,MMX,WKR,NQ,NP,NQ,NQ,NP)
01562          LP1= NQ + 1
01563          LP2= NQ + 1 + LP1
01564          LP3 = NQ + 1 + LP2
01565          LP4 = NQ + 1 + LP3
01566          LP5 = NQ + 1 + LP4
01567          LP6 = NQ + 1 + LP5
01570          LP7 = NQ + 1 + LP6
01571          LP8 = NQ*NQ + 1 + LP7
01572          CALL PSINV(WKR,WKRS,NQ,NQ,TOLENC,WORKST(1),WORKST(LP1),WORKST(LP2)
01572         *,WORKST(LP3),WORKST(LP4),WORKST(LP5),WORKST(LP6),WORKST(LP7))
01573          NT4=0
01574    510   NT2=0
01575          NT3=NT4+1
01576          NT4=NT3+19
01577          IF(NT4.GT.NQ)NT4=NQ
01601    520   NT1=NT2+1
01602          NT2=NT1+5
01603          IF(NT2.GT.NQ)NT2=NQ
01605          WRITE(6,3080)
01607   3080   FORMAT(1H///,5X,'COVARIENCE MATRIX OF BLE RESTRICTED BY THE HYPOTH
01607         *ESIS')
01610          WRITE(6,3018)(LP,IJ,IJ=NT1,NT2)
01617          II = NT1
01620          JJ = NT1 -1
01621          DO 530 I=NT3,NT4
01624          IF(II.NE.1)WRITE(6,3021)I
01630          IF(II.EQ.1)JJ = JJ + 1
01632          IF(JJ.GT.NT2)JJ=NT2
01634          IF(II.EQ.1)WRITE(6,3020)I,(WKRS(I,J),J=NT1,JJ)
01644          IF(II.EQ.1)II=II + 1
01646    530   CONTINUE
01650          IF(NT2.LT.NQ)GO TO 520
```

5

```
01652    688*           IF(NT4.LT.NW)GO TO 510
01652    689*     C*
01652    690*     C*    SQUARED MULTIPLE CORRELATION COEFFICIENT
01652    691*     C*
01654    692*           CALL MULTMX(MUX,CHV,ZWRK,NP,NQ,1,NP,NQ)
01655    693*           CALL MULTMX(ZTRP,ZWRK,Q1,1,NP,1,1 ,NP)
01656    694*           Q1 = Q1 - NPY
01657    695*           CALL MULTMX(HINV,ZWRK,UVEC,NQ,NP,1,NQ,NQ)
01660    696*           CALL MULTMX(CHV ,UVEC,Q2,1,NQ,1,1 ,NQ)
01661    697*           Q2 = Q2 - NPY
01662    698*           Q2 = (W-NPY) * Q2
01663    699*           R=Q1**2/Q2
01664    700*           WRITE(6,3093)
01666    701*      3093 FORMAT(/////,6X,'SQUARED MULTIPLE CORRELATION COEFFICIENT OF ULE R
01666    702*          *ESTRICTED BY THE HYPOTHESIS')
01667    703*           WRITE(6,3095)R
01672    704*      3095 FORMAT(//,T40,F15.9)
01673    705*           RETURN
01674    706*           END

      END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
@ FOR, GLMAOV,GLMAOV                                                     25 MAR 78        13:31:43.792
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E1201331DA)
THIS COMPILATION WAS DONE ON 25 MAR 78 AT 13:31:43

    MAIN PROGRAM

    STORAGE USED: CODE(1) 000551; DATA(0) 122167; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003    LMAOV
     0004    NRDU$
     0005    NIO2$
     0006    NWDU$
     0007    NSTOP$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0000    122076 101F        0001    000525 121G        0000    122066 15F         0000    122075 18F         0001    000531 19L
     0001    000534 30L         0001    000545 43L         0000 R 000000 ARRAY       0000 I 122027 I           0000 I 122031 LB
     0000 I 122051 LCHM        0000 I 122046 LCHV        0000 I 122053 LCI         0000 I 122050 LCM         0000 I 122040 LCV
     0000 I 122063 LEW         0000 I 122037 LGI         0000 I 122036 LGM         0000 I 122045 LGW         0000 I 122044 LHM
     0000 I 122041 LHV         0000 I 122022 LIMIT       0000 I 122054 LMX         0000 I 122062 LPP         0000 I 122060 LPS
     0000 I 122057 LPW         0000 I 122032 LR          0000 I 122033 LRC         0000 I 122034 LRI         0000 I 122047 LRW
     0000 I 122035 LTV         0000 I 122042 LW          0000 I 122064 LWAR        0000 I 122052 LWS         0000 I 122030 LX
     0000 I 122043 LXT         0000 I 122055 LZM         0000 I 122061 LZT         0000 I 122056 LZW         0000 I 122021 NALYS
     0000 I 122024 NM          0000 I 122025 NN          0000 I 122022 NP          0000 I 122023 NQ          0000 I 122065 NWS
     0000 R 122026 TOLENC


 00100      1*      C*     DRIVER PROGRAM GLMAOV WILL READ IN NO. OF ANALYSES TO BE TAKEN
 00100      2*      C*      AND DIMENSIONS OF MATH MODELS.  THEN IT CALLS SUBPROGRAM LMAOV
 00100      3*      C*      AFTER COMPUTING BASE ADDRESSES FROM LARGE WORKING ARRAY.
 00101      4*             DIMENSION ARRAY(42300)
 00103      5*             DATA LIMIT/42300/
 00105      6*             READ(5,15)NALYS,NP,NQ,NM,NN,TOLENC
 00115      7*       15    FORMAT(I3,1X,I2,1X,I2,1X,I2,1X,I2,1X,F10.8)
 00116      8*             IF(TOLENC.LE.0.)TOLENC = .0001
 00120      9*             DO 20 I=1,NALYS
 00123     10*             LX=NP+1
 00124     11*             LB=NP*NQ+LX+1
 00125     12*             LR=NQ+LB+1
 00126     13*             LRC=NM*NQ+1+LR
 00127     14*             LRI=NM+LRC+1
 00130     15*             LTV=NQ*NM+LRI+1
 00131     16*             LGM=NM+LTV+1
 00132     17*             LGI=NN*NQ+LGM+1
 00133     18*             LCV = NQ*NN + LGI + 1
 00134     19*             LHV = NQ + LCV + 1
 00135     20*             LW=NN+LHV+1
 00136     21*             LXT=NQ*NW+LW+1
 00137     22*             LHM=NQ*NP+LXT+1
 00140     23*             LGW=NN*NQ+LHM+1
 00141     24*             LCHV=NN*NN+LGW+1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
00142     25*          LRW=NN+LCHV+1
00143     26*          LCM=NM*NM+LRW+1
00144     27*          LCHM=NQ*NQ+LCM+1
00145     28*          LWS=NN+NM+LCHM+1
00146     29*          LCI = NQ*NQ + LWS + 1
00147     30*          LHX=NQ*NQ+LCI+1
00150     31*          LZM=NP*NQ+LHX+1
00151     32*          LZW=NP+LZM+1
00152     33*          LPW=NP+LZW+1
00153     34*          LPS=NQ*NP+LPW+1
00154     35*          LZT=NP*NP+LPS+1
00155     36*          LPP=NP+LZT+1
00156     37*          LEW = NP * NP + LPP + 1
00157     38*          LWAR= NQ*NQ + LEW + 1
00160     39*          NWS = NN*NW + 1 + LWAR
00161     40*          IF(NWS.GE.LIMIT)GO TO 30
00163     41*          CALL LMAOV(ARRAY(1),ARRAY(LX),ARRAY(LW),ARRAY(LR),ARRAY(LKC),
00163     42*         *ARRAY(LRI),ARRAY(LTV),ARRAY(LGM),ARRAY(LGI),ARRAY(LGI),ARRAY(LCV),
00163     43*         *ARRAY(LHV),ARRAY(LW),ARRAY(LXT),ARRAY(LHM),ARRAY(LGW),ARRAY(LCHV),
00163     44*         *ARRAY(LRW),ARRAY(LCM),ARRAY(LCHM),ARRAY(LWS),ARRAY(LCI),ARRAY(LHX)
00163     45*         *,ARRAY(LZM),ARRAY(LZW),ARRAY(LPW),ARRAY(LPW),ARRAY(LPS),ARRAY(LZT)
00163     46*         *,ARRAY(LPP),ARRAY(LEW),ARRAY(LWAR),ARRAY(NWS),
00163     47*         *NP,NQ,NM,NN,I,TOLENC,NWS)
00164     48*          IF(I.EQ.NALYS)GO TO 19
00166     49*          READ(5,18)NN
00171     50*      18  FORMAT(I2)
00172     51*      19  CONTINUE
00173     52*      20  CONTINUE
00175     53*          GO TO 40
00176     54*      30  LEW = LIMIT - NWS
00177     55*          WRITE(6,101)LEW
00202     56*     101  FORMAT(5X,'**** DIMENSIONS OF MATH MODELS ARE TOO LARGE FOR USE
00202     57*         *OF THIS PROGRAM *** STORAGE CAPICITY OF LMAOV IS EXCEEDED**** ANA
00202     58*         *LYSIS ASKS FOR',I5,' MORE LOCATIONS THAN AVAILABLE')
00203     59*      40  CONTINUE
00204     60*          STOP
00205     61*          END

       END OF COMPILATION:        NO DIAGNOSTICS.
```